**《信息安全基础》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | **孙莹莹** | | **年级** | | **2020级** |
| **学号** | | **20204104** | | **专业、班级** | | **计算机科学与技术卓越一班** |
| **实验名称** | **加解密算法的实现** | | | | | |
| **实验时间** | **2023.4.9** | | **实验地点** | | **DS3402** | |
| **实验成绩** |  | | **实验性质** | | **□验证性 □设计性 ☑综合性** | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  评语：  评价教师签名（电子签名）： | | | | | | |
| 一、实验目的  掌握频度分析法原理和Feistel加解密原理 | | | | | | |
| 二、实验项目内容  1.使用频度分析法解密以下文本，并给出替换表：  UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ  2.编程实现Feistel加密解密以下文本：  CQUINFORMATIONSECURITYEXP | | | | | | |
| 三、实验设计（实验原理、真值表、原理图等）  **1.频度分析原理**  对于任何一种书面语言而言，不同的字母或字母组合出现的频率各不相同。如果以这种语言书写足够长的文本，都呈现出大致相同的特征字母分布规律，如下表所示：  表格  描述已自动生成  在上表中，不少字母出现的概率近乎相等，但也有极少数字母出现的概率有较大差异。为了分析方便密文信息，常将英文字母表按字母出现的概率大小分类，分类情况如下：  --------------------------  极高频 E  次高频 T A O I N S H R  中等频 D L  低频 C U M W F G Y P B  甚低频 V K J X Q Z  --------------------------  语言的单字母统计特性没有反映出英文双字母和多字母的特征，在双字母中统计出概率最大的30对字母按概率大小排列为：  th he in er an re ed on es st en at to nt ha nd ou ea ng as or ti is et it ar te se hi of  类似的，我们还可以考察英文课文中三字母出现的频率。按Beker在1982年统计的结果(样本总数100 360)得到概率最大的20组三字母按概率大小排列为：  the ing and her ere ent tha nth was eth for dth hat she ionhis sth ers ver  特别地，the出现的频率几乎为ing的3倍。  **2.Feistel网络原理**    加密过程：  Li = Ri-1  Ri = Li-1 ⊕f(Ri-1,Ki)  解密过程：  显然Ri-1= Li 直接获得。  而Li-1 通过下式可得：  Li-1 = Li-1 ⊕ f (Ri-1,Ki) ⊕ f (Ri-1,Ki)  = Ri ⊕ f (Li,Ki) | | | | | | |
| 四、实验过程或算法(关键步骤、核心代码注解等）  **1.使用频度分析法解密文本**  首先我们使用编程实现密文各个字母的频率统计，代码如下：   1. **using** **namespace** std; 2. **char** should[26] = {'E', 'T', 'A', 'O', 'N', 'I', 'S', 'R', 'H', 'L', 'D', 'C', 'U', 'P', 'F', 'M', 'W', 'Y', 'B', 'G', 'V', 'K', 'Q', 'X', 'J', 'Z'}; 4. **bool** cmp(pair<**char**, **int**> a, pair<**char**, **int**> b) 5. { 6. **return** a.second > b.second; 7. } 9. **int** main() 10. { 11. string code = "UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ"; 12. map<**char**, **int**> charFrequency; 13. map<**char**, **char**> table; 15. **for** (**int** i = 0; i < code.length(); i++) 16. { 17. **if** (code[i] >= 'A' && code[i] <= 'Z') 18. { 19. charFrequency[code[i]]++; 20. } 21. } 22. vector<pair<**char**, **int**>> vec; 23. **for** (map<**char**, **int**>::iterator it = charFrequency.begin(); it != charFrequency.end(); it++) 24. { 25. vec.push\_back(pair<**char**, **int**>(it->first, it->second)); 26. } 27. sort(vec.begin(), vec.end(), cmp); 29. system("pause"); 30. **return** 0; 31. }   统计出来的字符频率如下：    采用频度分析法，根据以下字符出现频度，先假设出现次数最多的P代表的原文是极高频的E，Z代表是的T，依次往下按照频率替代，但得到的结果不甚满意。    采用社会工程学分析，保留P->E, Z->T这两个对照表，替换得到的结果如下：  **Ut** QSO VUOHXMOeV GeOteEVSG tWSt OeFeESX UDBMETSX AIt VUEeHt HMDtSHtO WSFe AeeD TSVe QUtW YMXUtUHSX EeYEeOeDtStUFeO MB **tWe** FUet HMDJ UD TMOHMQ  根据英文知识 tWe很可能对应the,即W->H，Ut放在句子开头可能为It,即U->I（这个假设也是符合频度的，因为密文中U出现了10次，属于较高频次，而再明文中I也是属于次高频一类的），此时再次验证假设，得到的句子如下：  it QSO ViOHXMOeV GeOteEVSG thSt OeFeESX iDBMETSX AIt ViEeHt HMDtSHtO hSFe AeeD TSVe Qith YMXitiHSX EeYEeOeDtStiFeO MB the Fiet HMDJ iD TMOHMQ  此时密文中出现10次以上的字母只有S没有找到对应明文，那么我们循环遍历，让S等于次高频中的某个字母，打印出句子结果，来确定S对应的明文。  比如让S对应‘o’:  it QoO ViOHXMOeV GeOteEVoG thot OeFeEoX iDBMEToX AIt ViEeHt HMDtoHtO hoFe AeeD ToVe Qith YMXitiHoX EeYEeOeDtotiFeO MB the Fiet HMDJ iD TMOHMQ  可以发现thot这个单词是不存在的，这里启示我们这个单词应当是that,因此S对应的是a，替换后句子如下：  it QaO ViOHXMOeV GeOteEVaG that OeFeEaX iDBMETaX AIt ViEeHt HMDtaHtO haFe AeeD TaVe Qith YMXitiHaX EeYEeOeDtatiFeO MB the Fiet HMDJ iD TMOHMQ  私以为O出现了九次应当属于中频，但替换后无法得到合适的结果，因此再次采用遍历，让O等于次高频中的某个字母，打印出句子结果，来确定S对应的明文。最后得到O对应‘s’时比较合适，对于M的查找同理，替换后句子如下：  it Qas VisHXoseV GesteEVaG that seFeEaX iDBoETaX AIt ViEeHt HoDtaHts haFe AeeD TaVe Qith YoXitiHaX EeYEeseDtatiFes oB the Fiet HoDJ iD TosHoQ  根据Qas 和Qith猜测Q->W,那么可以大致确定出现次数为3次及以下的应当属于低频和甚低频这一类，找到oB这个关键词，带入低频及甚低频中的字母，发现只有f符合要求，因此B->F:  it was VisHXoseV GesteEVaG that seFeEaX iDfoETaX AIt ViEeHt HoDtaHts haFe AeeD TaVe with YoXitiHaX EeYEeseDtatiFes of the Fiet HoDJ iD TosHow  根据iD推测D应当为n,此时确定出现次数大于6次的应当属于次高频一类的字母，而出现次数为5应当属于中等频，而V和X都出现五次，分别带入D、L:  it was disHlosed GesteEdaG that seFeEal infoETal AIt diEeHt HontaHts haFe Aeen Tade with YolitiHal EeYEesentatiFes of the Fiet HonJ in TosHow  根据disHlosed推测H->C, Aeen推测A->B, bIt推测I->U，Tade推测T->M, diEect推测E->R, seFeral推测F->v,Yolitical推测Y->P,GesterdaG推测G->Y。（上述推测都是结合已推测出来的字母的词频和密文字母的词频所推测的，除了少部分不符合词频的字母外，此处不再赘述，可见实验结果分析一栏）  **2.编程实现Feistel加密解密以下文本**  CQUINFORMATIONSECURITYEXP  **（1）将字符转换为二进制编码**  因为在Feistel中需要用到异或算法，所以需要先把文本编码为二进制字符才能进行异或。因此把每个字符对应到它的ASCII码，然后转换为二进制，之后将所有字符的编码连接起来，就得到了明文。对于解密后的密文，采用逆方法则得到对应的字符。   1. information = "CQUINFORMATIONSECURITYEXP" 2. #将原文转为二进制，便于后面异或运算 3. charAs = [format(ord(a),'08b') **for** a **in** information] 4. codeInfor = ''.join(charAs) 5. #还原回原来的字符串 6. wordsRead = ''.join([chr(a) **for** a **in** [int(i,2) **for** i **in** [decodeWords[a:a+8] **for** a **in** range(0,len(decodeWords),8)]]])   **（2）根据密钥和原文加密**  在encode函数里，我们传入需要加密的data和密钥key,取L为data的左半部分，R为data的右半部分，然后循环rounds(16)轮次。在每一轮更新L和R，L等于上一轮的R，而R则等于上一轮的L和f(R,key)的结果异或。在这里设置f函数进行的也是异或运算。每一轮计算完成后更新key,更新采用的是将key右移一位,得到下一次的key。最后返回密文   1. rounds = 16 3. **def** fun(R,key): 4. **return** int(R,2) ^ int(key,2) 6. **def** encode(data,key): 7. L = data[:len(data)//2] 8. R = data[len(data)//2:] 9. **for** i **in** range(rounds): 10. tmp = R; 11. R = int(L, 2) ^ fun(R, key) 12. R = format(R,f'0{len(tmp)}b') 13. L = tmp 14. key = key[15] + key[:15] #移位更新key 15. **return** L+R 16. **根据密钥和密文解密**   在decode函数里，我们传入密文data和密钥key,取L为data的左半部分，R为data的右半部分，然后循环rounds(16)轮次。在每一轮更新L和R，首先获得该轮次的密钥，即key左移一位。R等于上一轮的L，而L则等于上一轮的R和f(L,key)的结果异或。最后返回明文。   1. **def** decode(data,key): 2. L = data[:len(data) // 2] 3. R = data[len(data) // 2:] 4. **for** i **in** range(rounds): 5. key = key[1:] + key[0] 6. tmp = L 7. L = int(R,2) ^ fun(L,key) 8. L = format(L,f'0{len(tmp)}b') 9. R = tmp; 10. **return** L + R 11. **main()函数**   main函数则是对上述过程的调用，首先将信息进行编码，设置密钥k=0000000000000001,之后调用encode和decode函数进行加密解密，最后返回解密后的信息。   1. **if** \_\_name\_\_ == '\_\_main\_\_': 2. information = "CQUINFORMATIONSECURITYEXP" 3. charAs = [format(ord(a),'08b') **for** a **in** information] 4. codeInfor = ''.join(charAs) #将原文转为二进制，便于后面异或运算 5. keyK = "0000000000000001" 6. #加密 7. codedWords = encode(codeInfor,keyK) 8. # 解密 9. decodeWords = decode(codedWords,keyK) 10. #还原回原来的字符串 11. wordsRead = ''.join([chr(a) **for** a **in** [int(i,2) **for** i **in** [decodeWords[a:a+8] **for** a **in** range(0,len(decodeWords),8)]]]) 12. **print**("The original information is: ",information) 13. **print**("The encoded information is: ", codedWords) 14. **print**("The deencoded information is: ", decodeWords) 15. **print**("The readable deencoded information is: ", wordsRead) | | | | | | |
| 五、实验过程中遇到的问题及解决情况(主要问题及解决情况)  **1.使用频度分析法解密文本**  分析中的难点主要在于几个在密文中出现的频次和明文中不大对应的字母，比如H->C, H在密文中出现了7次，而如果根据出现了6次的D->N推断的话（N是次高频的字母），那么7>6应当也会是H对应的字母应当是次高频的，但实际上C是一个低频的字母。 如果单纯的按密文频次统计然后根据次数对应到相应频次的明文，是无法得到正确的答案范围和最终结果的。所以还需要结合社会工程学分析。  **2.编程实现Feistel加密解密文本**  一开始在编写代码时没有加入17行，则会产生报错：   1. R = int(L, 2) ^ fun(R, key) 2. R = format(R,f'0{len(tmp)}b')     int()函数用来将一个字符串或数字类型转换成整数，如果只有一个参数值，这个值可以是字符串或数字；但是传入两个参数，第一个参数是字符串，第二个参数是进制（二进制、八进制、十进制或十六进制）。R在进行异或运算后，类型是int,那么第一个参数传入的是数字，第二个是二进制，所以会报错。应当要把R再次转换为字符串类型才可以正常运行。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  **1.使用频度分析法解密文本**  最终得到的密码表如下，为使显示最终句子好看故全部改为了小写字母。    按词频排序版： 按字母排序版：  分析得到的原文如下：  it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow   1. **编程实现Feistel加密解密文本**   原文为：CQUINFORMATIONSECURITYEXP  密文为：  输出结果如下： | | | | | | |